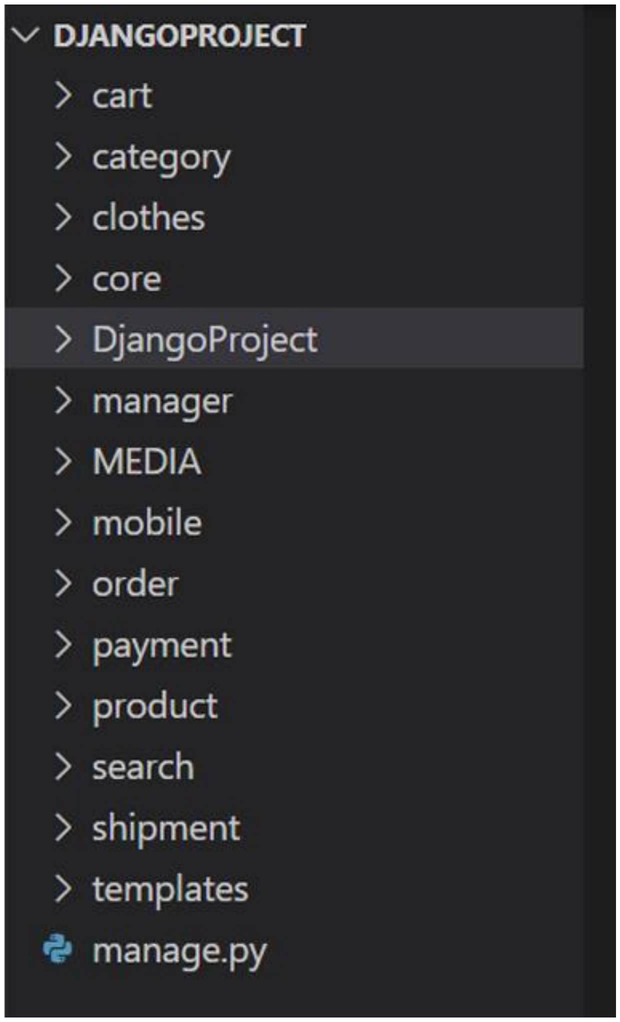
Họ tên: Lê Duy Mạnh

MSV: B20DCCN423

I. Tạo 10 services

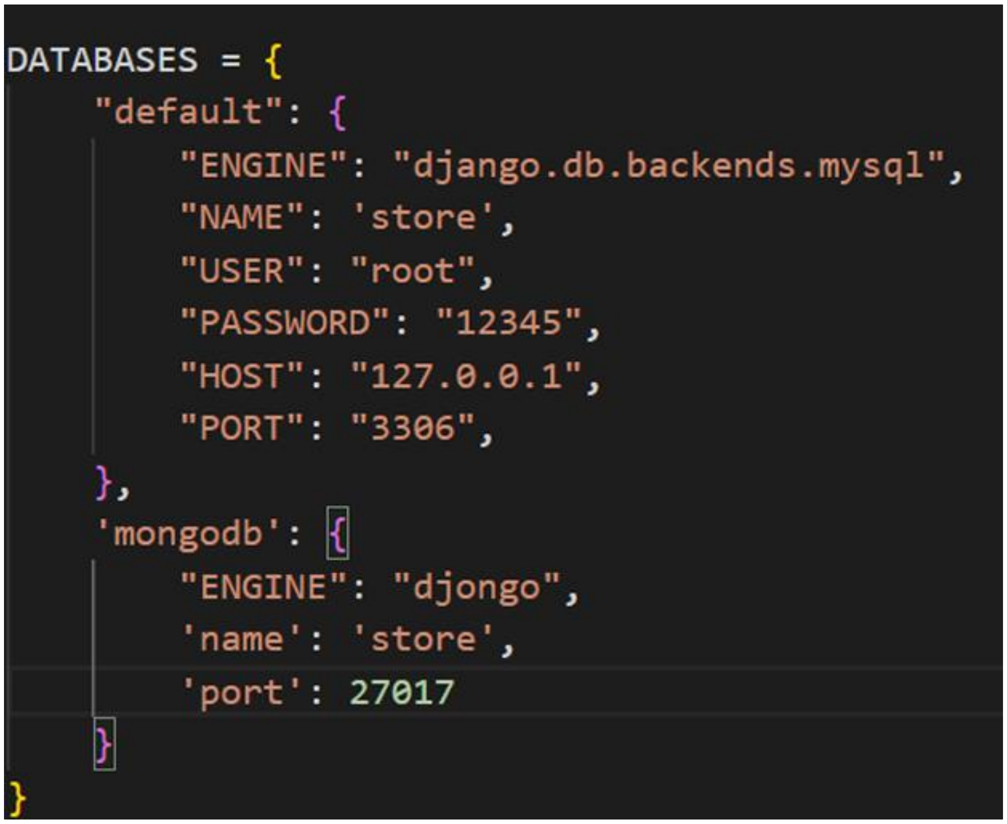
Tạo các app service theo lệnh

o User: django-admin startapp user
o Product: django-admin startapp product
o Book: Django-admin startapp book
o Cloth: Django-admin startapp cloth
o Mobile: Django-admin startapp mobile
o Category: Django-admin startapp category
o Cart: Django-admin startapp cart
o Search: Django-admin startapp search
o Payment: Django-admin startapp payment
o Shipment: Django-admin startapp shipment

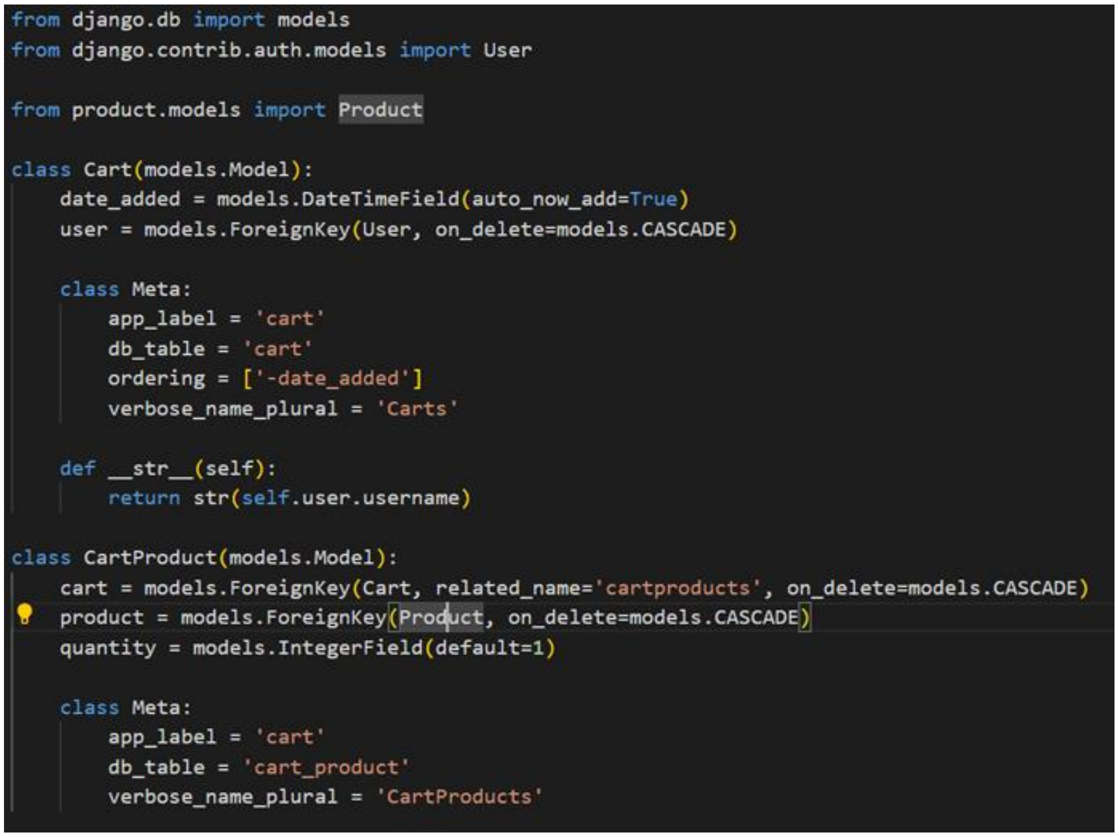


```
DJANGOPROJECT
  cart
  category
  clothes
  core
  DjangoProject
  manager
  MEDIA
  mobile
  order
  payment
  product
  search
  shipment
  templates
  manage.py
```

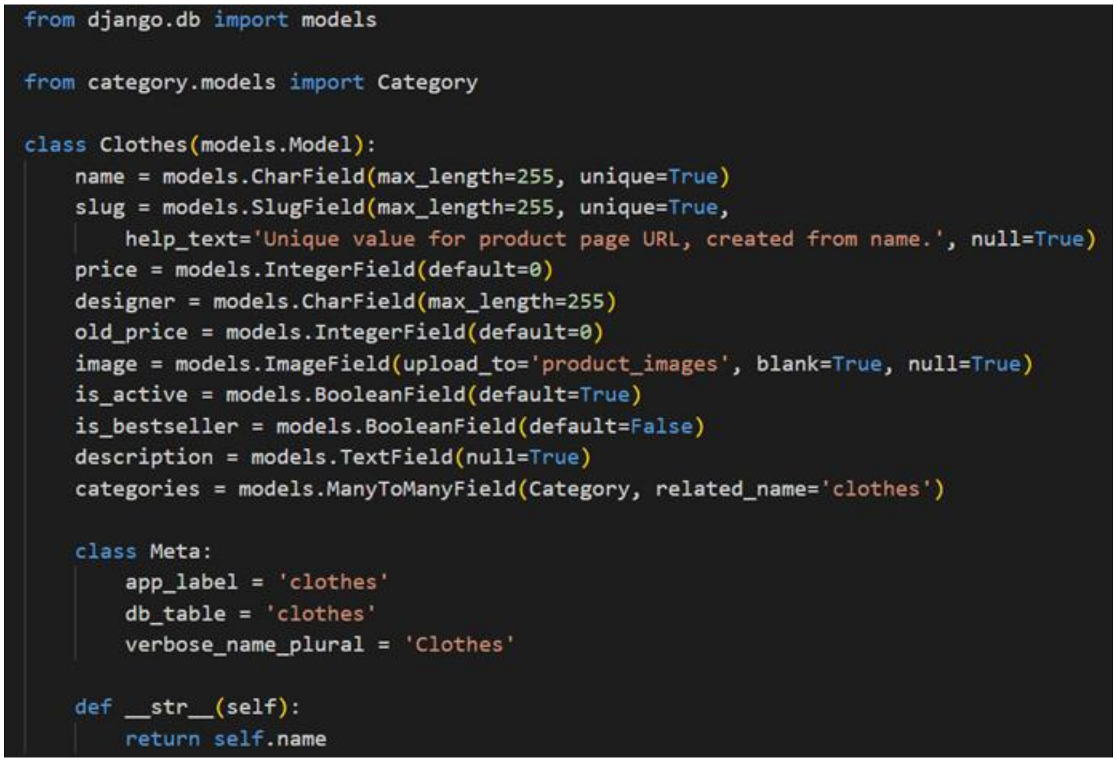
Thêm các app vào INSTALLED_APPS trong file setting.py Thiết lập DATABASES trong file setting.py


```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "NAME": 'store',
        "USER": "root",
        "PASSWORD": "12345",
        "HOST": "127.0.0.1",
        "PORT": "3306",
    },
    'mongodb': {
        "ENGINE": "djongo",
        'name': 'store',
        'port': 27017
    }
}
```

App Cart

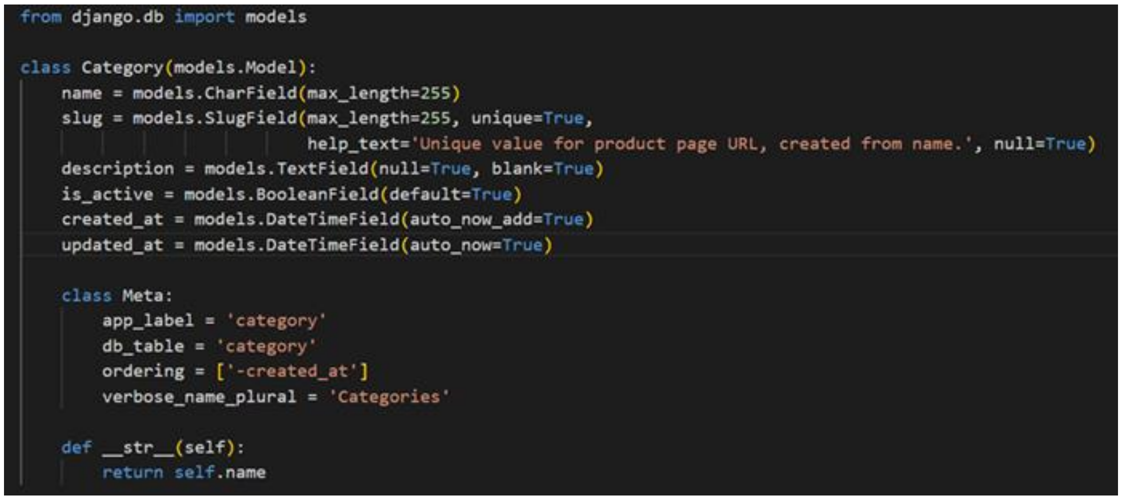

```
from django.db import models
from django.contrib.auth.models import User

from product.models import Product

class Cart(models.Model):
    date_added = models.DateTimeField(auto_now_add=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    class Meta:
        app_label = 'cart'
        db_table = 'cart'
        ordering = ['-date_added']
        verbose_name_plural = 'Carts'

    def __str__(self):
        return str(self.user.username)

class CartProduct(models.Model):
    cart = models.ForeignKey(Cart, related_name='cartproducts', on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=1)

    class Meta:
        app_label = 'cart'
        db_table = 'cart_product'
        verbose_name_plural = 'CartProducts'
```

App clothes



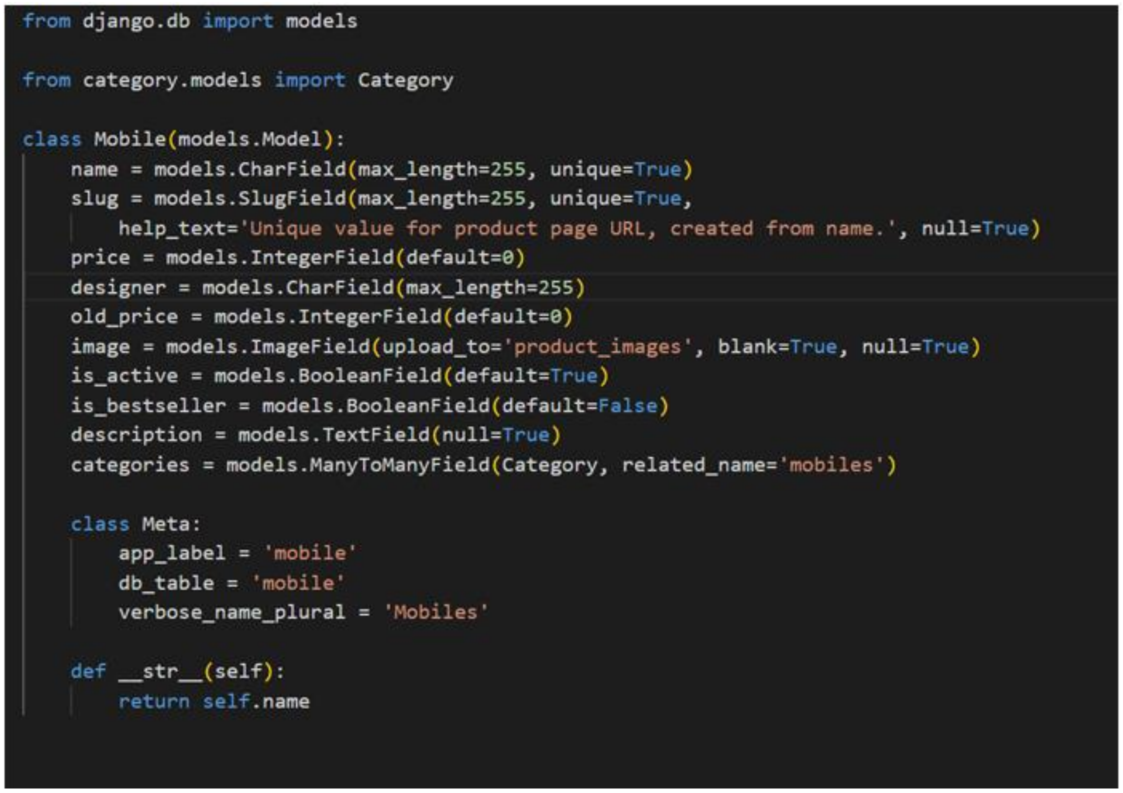
```python
from django.db import models

from category.models import Category

class Clothes(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
        help_text='Unique value for product page URL, created from name.', null=True)
    price = models.IntegerField(default=0)
    designer = models.CharField(max_length=255)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    categories = models.ManyToManyField(Category, related_name='clothes')

    class Meta:
        app_label = 'clothes'
        db_table = 'clothes'
        verbose_name_plural = 'Clothes'

    def __str__(self):
        return self.name
```

App category



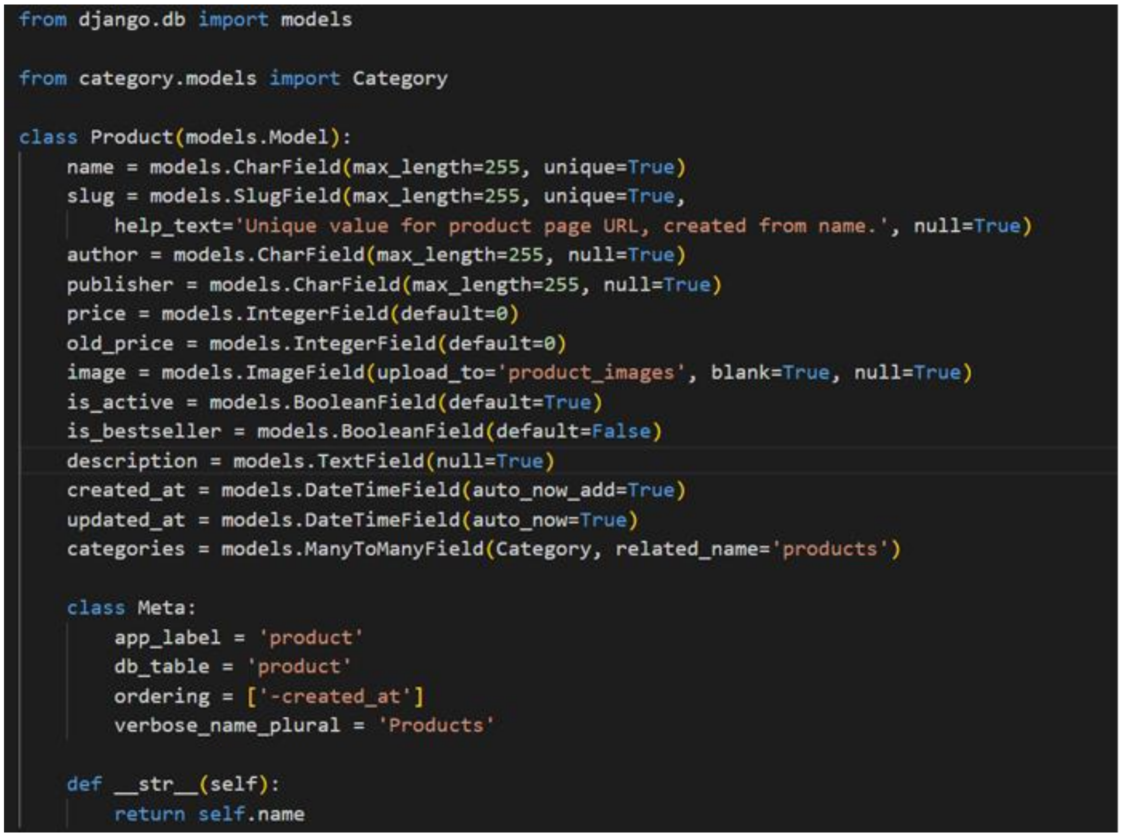
```python
from django.db import models

class Category(models.Model):
    name = models.CharField(max_length=255)
    slug = models.SlugField(max_length=255, unique=True,
                            help_text='Unique value for product page URL, created from name.', null=True)
    description = models.TextField(null=True, blank=True)
    is_active = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        app_label = 'category'
        db_table = 'category'
        ordering = ['-created_at']
        verbose_name_plural = 'Categories'

    def __str__(self):
        return self.name
```

app mobile



```python
from django.db import models

from category.models import Category

class Mobile(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
        help_text='Unique value for product page URL, created from name.', null=True)
    price = models.IntegerField(default=0)
    designer = models.CharField(max_length=255)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    categories = models.ManyToManyField(Category, related_name='mobiles')

    class Meta:
        app_label = 'mobile'
        db_table = 'mobile'
        verbose_name_plural = 'Mobiles'

    def __str__(self):
        return self.name
```

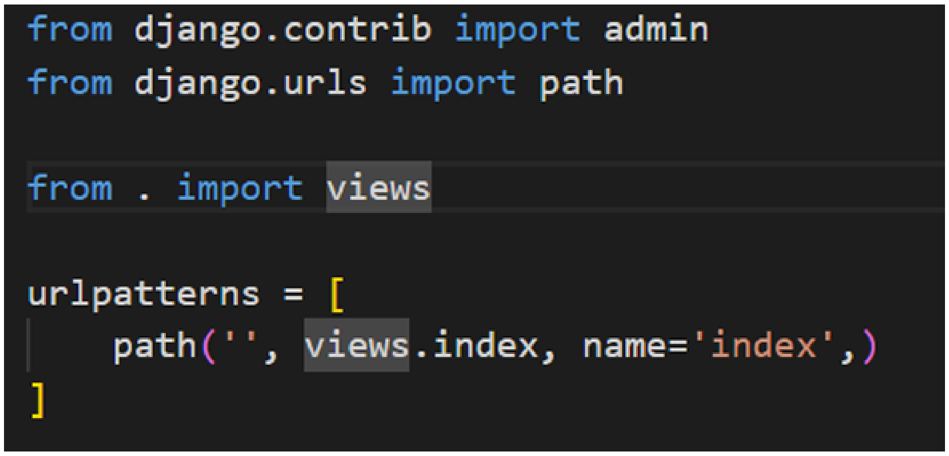
App Product



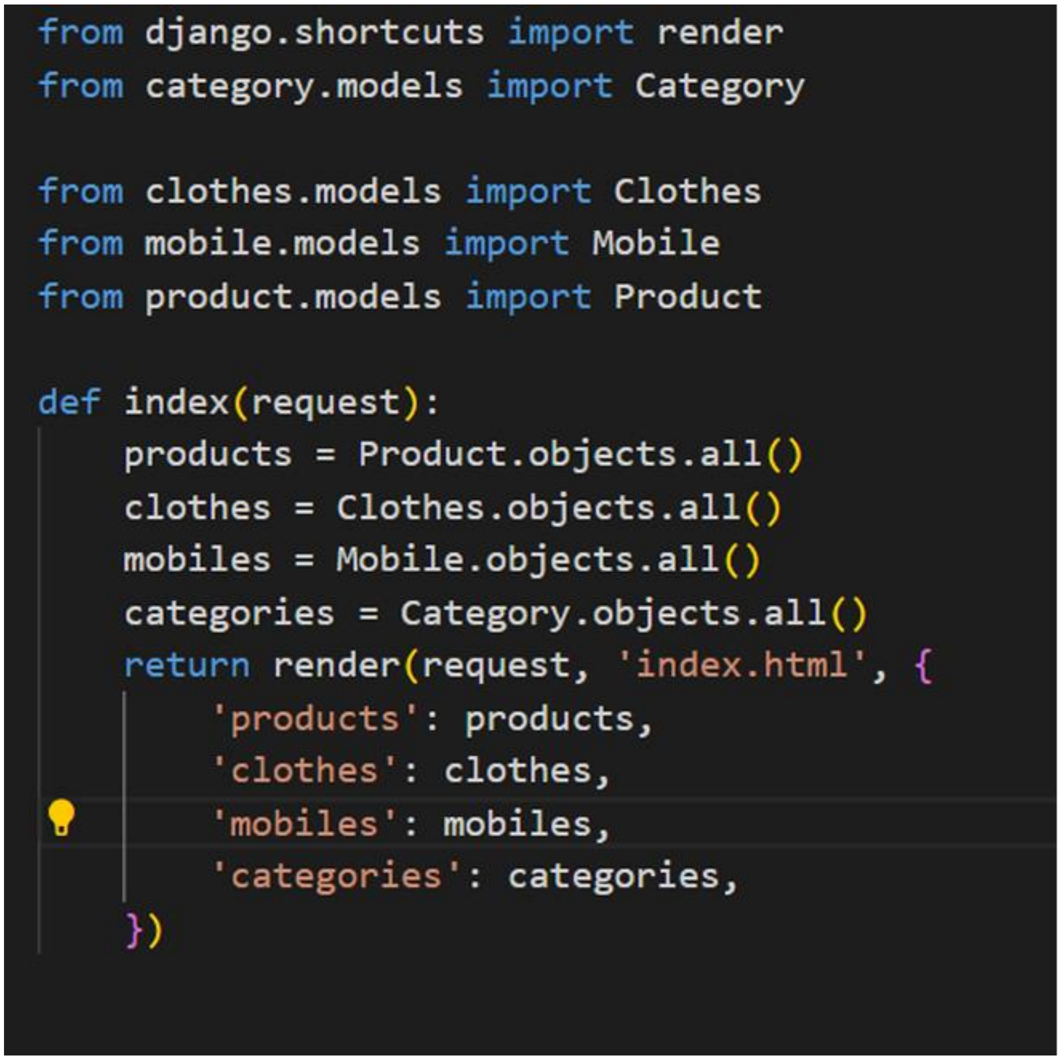
```python
from django.db import models

from category.models import Category

class Product(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
        help_text='Unique value for product page URL, created from name.', null=True)
    author = models.CharField(max_length=255, null=True)
    publisher = models.CharField(max_length=255, null=True)
    price = models.IntegerField(default=0)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    categories = models.ManyToManyField(Category, related_name='products')

    class Meta:
        app_label = 'product'
        db_table = 'product'
        ordering = ['-created_at']
        verbose_name_plural = 'Products'

    def __str__(self):
        return self.name
```

II. Tạo Rest API để kết nối với các dịch vụ

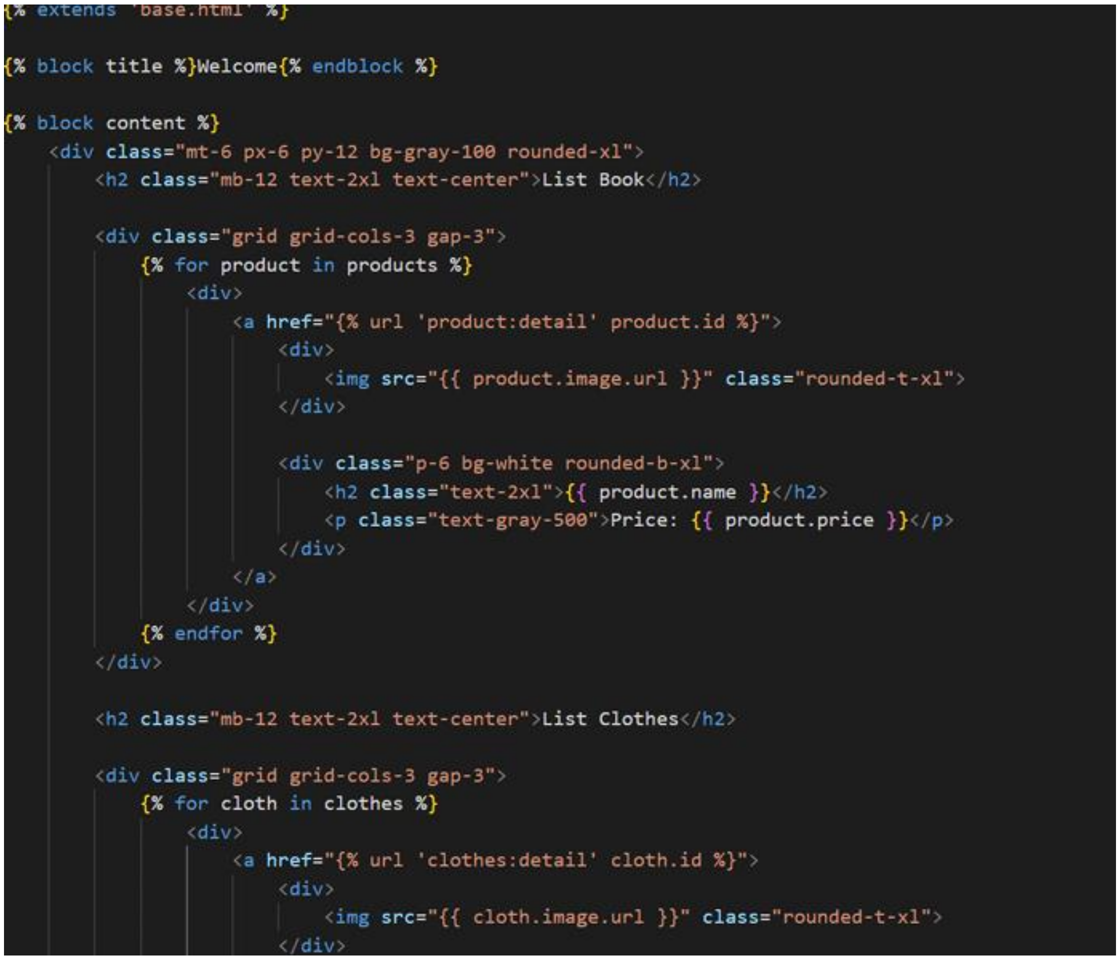
Hiển thị tất cả các product, clothes, mobile các file urls trong từng app tương ứng



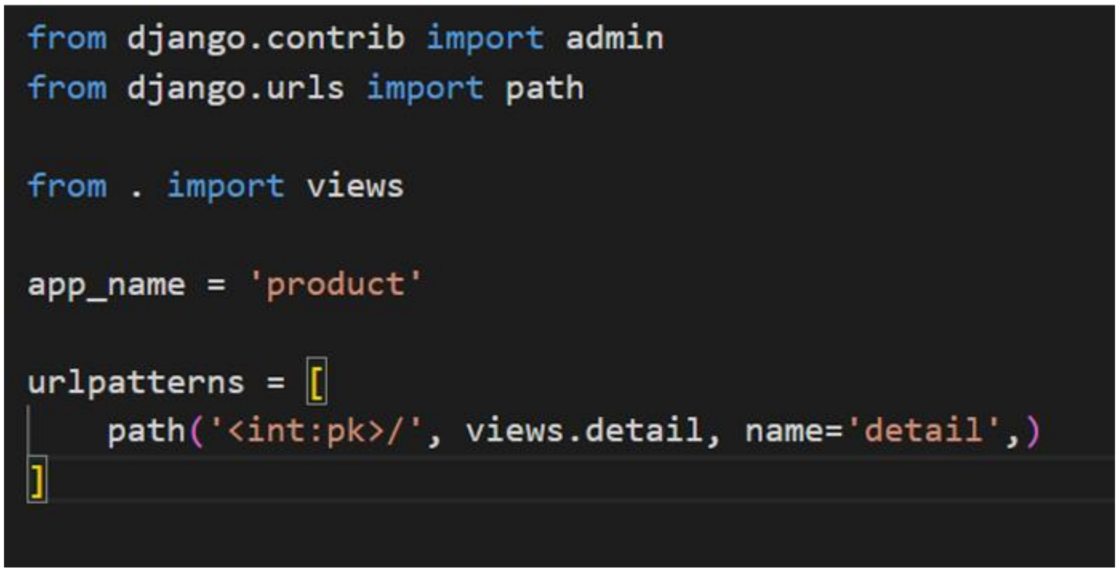
```python
from django.contrib import admin
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index',)
]
```



```python
from django.shortcuts import render
from category.models import Category

from clothes.models import Clothes
from mobile.models import Mobile
from product.models import Product

def index(request):
    products = Product.objects.all()
    clothes = Clothes.objects.all()
    mobiles = Mobile.objects.all()
    categories = Category.objects.all()
    return render(request, 'index.html', {
        'products': products,
        'clothes': clothes,
        'mobiles': mobiles,
        'categories': categories,
    })
```



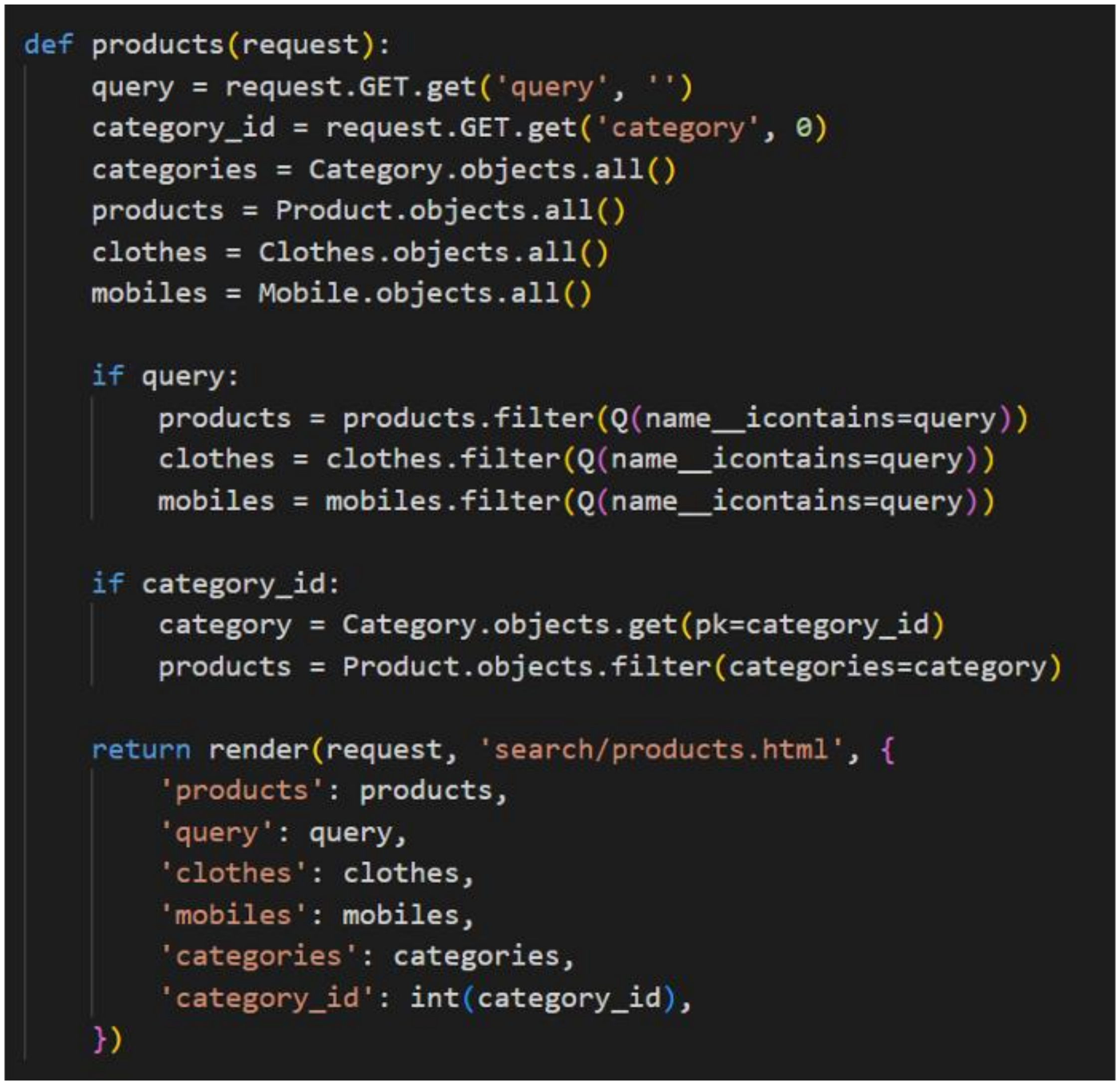
```
{% extends 'base.html' %}

{% block title %}Welcome{% endblock %}

{% block content %}
    <div class="mt-6 px-6 py-12 bg-gray-100 rounded-xl">
        <h2 class="mb-12 text-2xl text-center">List Book</h2>

        <div class="grid grid-cols-3 gap-3">
            {% for product in products %}
                <div>
                    <a href="{% url 'product:detail' product.id %}">
                        <div>
                            <img src="{{ product.image.url }}" class="rounded-t-xl">
                        </div>

                        <div class="p-6 bg-white rounded-b-xl">
                            <h2 class="text-2xl">{{ product.name }}</h2>
                            <p class="text-gray-500">Price: {{ product.price }}</p>
                        </div>
                    </a>
                </div>
            {% endfor %}
        </div>

        <h2 class="mb-12 text-2xl text-center">List Clothes</h2>

        <div class="grid grid-cols-3 gap-3">
            {% for cloth in clothes %}
                <div>
                    <a href="{% url 'clothes:detail' cloth.id %}">
                        <div>
                            <img src="{{ cloth.image.url }}" class="rounded-t-xl">
                        </div>
```



```
from django.contrib import admin
from django.urls import path

from . import views

app_name = 'product'

urlpatterns = [
    path('<int:pk>/', views.detail, name='detail',)
]
```

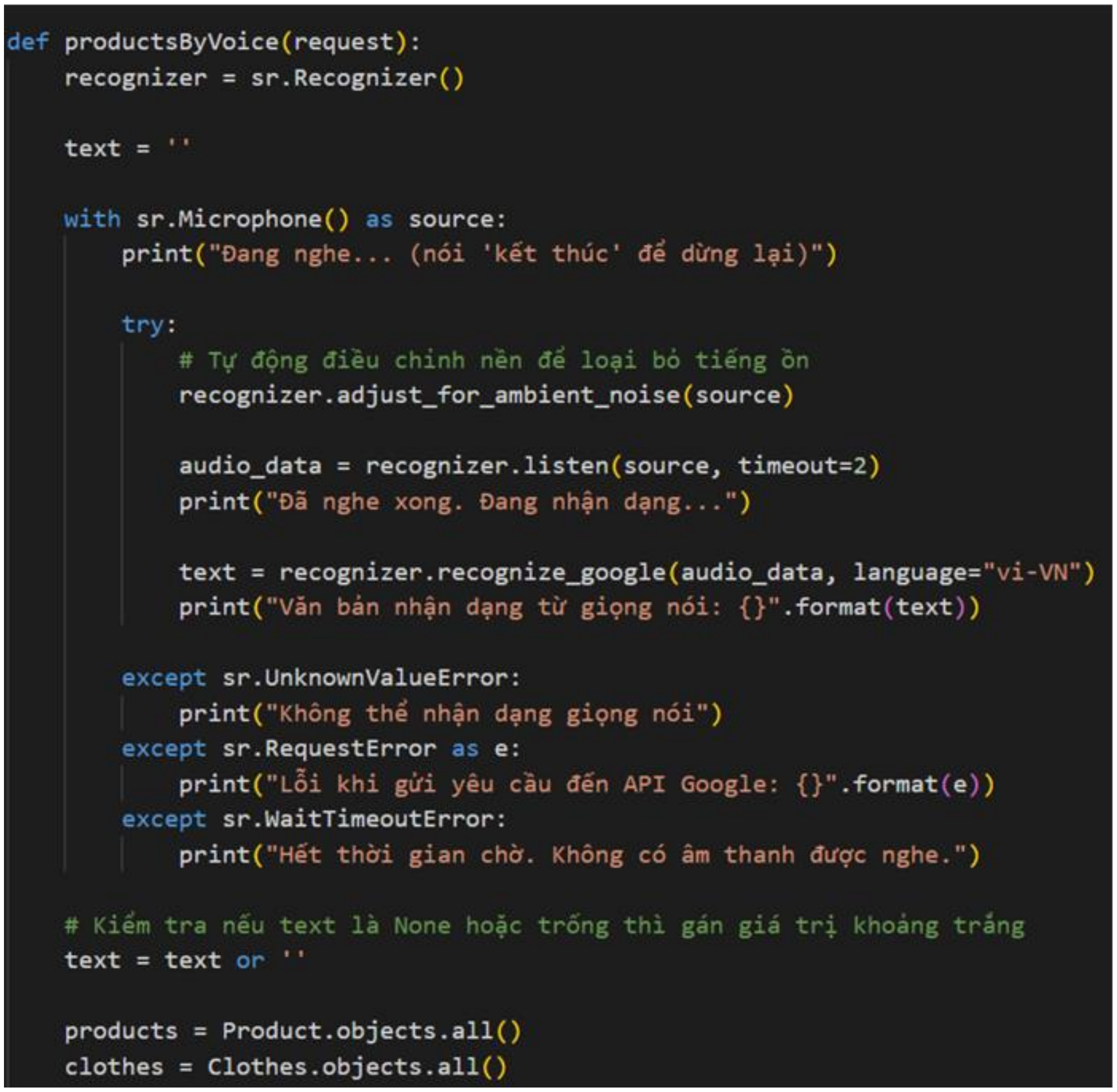
III. Search

quy trình tại file view.py trong app search đối với từ khóa nhập vào

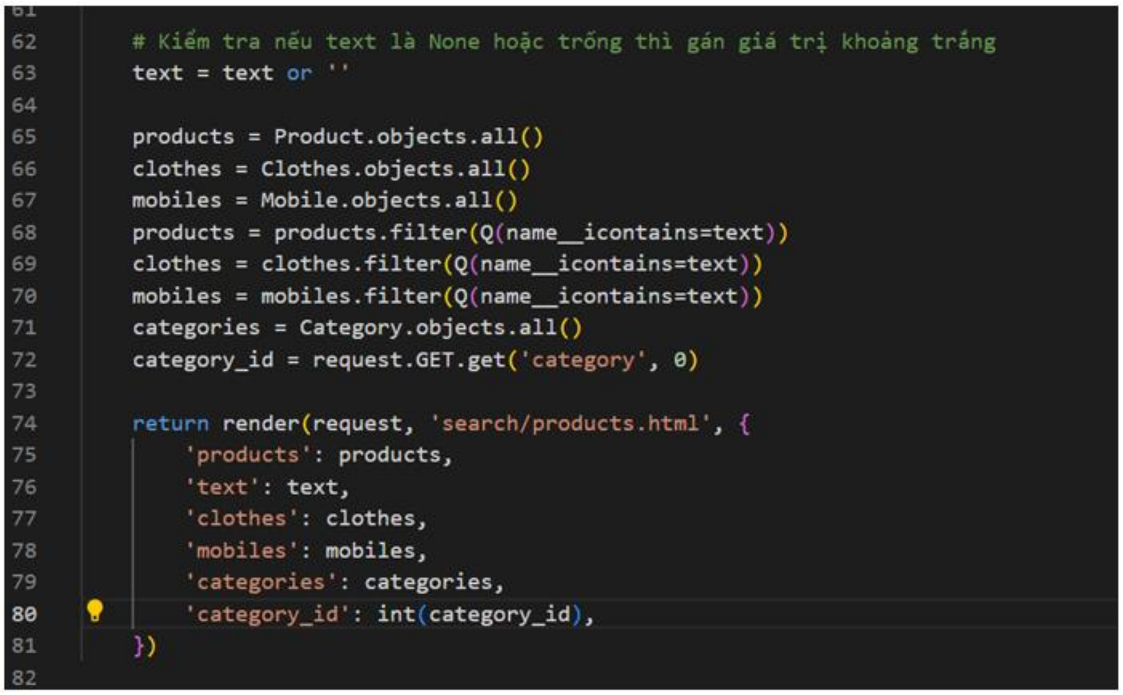


```python
def products(request):
    query = request.GET.get('query', '')
    category_id = request.GET.get('category', 0)
    categories = Category.objects.all()
    products = Product.objects.all()
    clothes = Clothes.objects.all()
    mobiles = Mobile.objects.all()

    if query:
        products = products.filter(Q(name__icontains=query))
        clothes = clothes.filter(Q(name__icontains=query))
        mobiles = mobiles.filter(Q(name__icontains=query))

    if category_id:
        category = Category.objects.get(pk=category_id)
        products = Product.objects.filter(categories=category)

    return render(request, 'search/products.html', {
        'products': products,
        'query': query,
        'clothes': clothes,
        'mobiles': mobiles,
        'categories': categories,
        'category_id': int(category_id),
    })
```

đối với xử lí tìm kiếm giọng nói

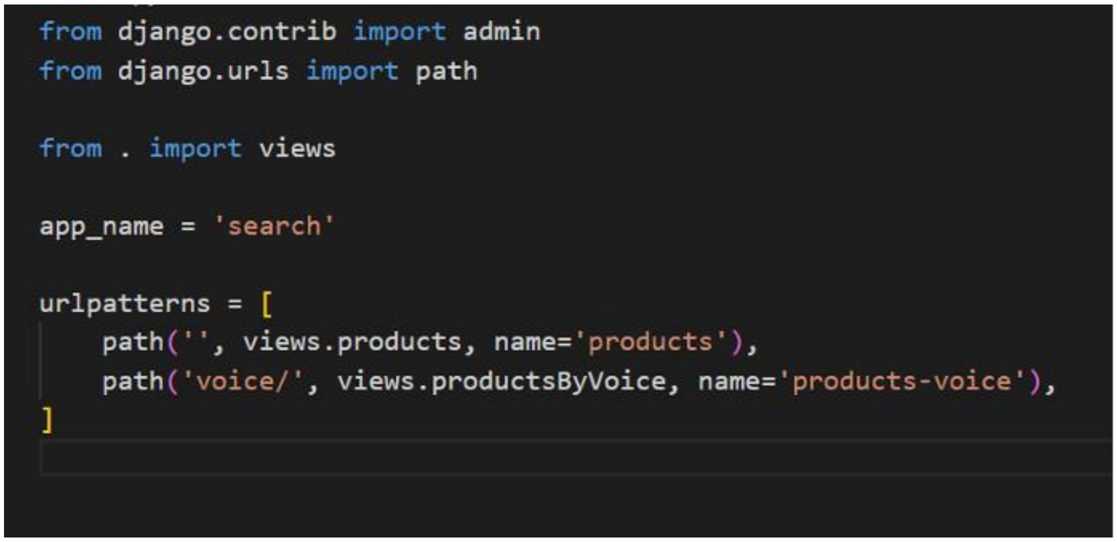


```python
def productsByVoice(request):
    recognizer = sr.Recognizer()

    text = ''

    with sr.Microphone() as source:
        print("Đang nghe... (nói 'kết thúc' để dừng lại)")

        try:
            # Tự động điều chỉnh nền để loại bỏ tiếng ồn
            recognizer.adjust_for_ambient_noise(source)

            audio_data = recognizer.listen(source, timeout=2)
            print("Đã nghe xong. Đang nhận dạng...")

            text = recognizer.recognize_google(audio_data, language="vi-VN")
            print("Văn bản nhận dạng từ giọng nói: {}".format(text))

        except sr.UnknownValueError:
            print("Không thể nhận dạng giọng nói")
        except sr.RequestError as e:
            print("Lỗi khi gửi yêu cầu đến API Google: {}".format(e))
        except sr.WaitTimeoutError:
            print("Hết thời gian chờ. Không có âm thanh được nghe.")

    # Kiểm tra nếu text là None hoặc trống thì gán giá trị khoảng trắng
    text = text or ''

    products = Product.objects.all()
    clothes = Clothes.objects.all()
```



```
    # Kiểm tra nếu text là None hoặc trống thì gán giá trị khoảng trắng
    text = text or ''

    products = Product.objects.all()
    clothes = Clothes.objects.all()
    mobiles = Mobile.objects.all()
    products = products.filter(Q(name__icontains=text))
    clothes = clothes.filter(Q(name__icontains=text))
    mobiles = mobiles.filter(Q(name__icontains=text))
    categories = Category.objects.all()
    category_id = request.GET.get('category', 0)

    return render(request, 'search/products.html', {
        'products': products,
        'text': text,
        'clothes': clothes,
        'mobiles': mobiles,
        'categories': categories,
        'category_id': int(category_id),
    })
```

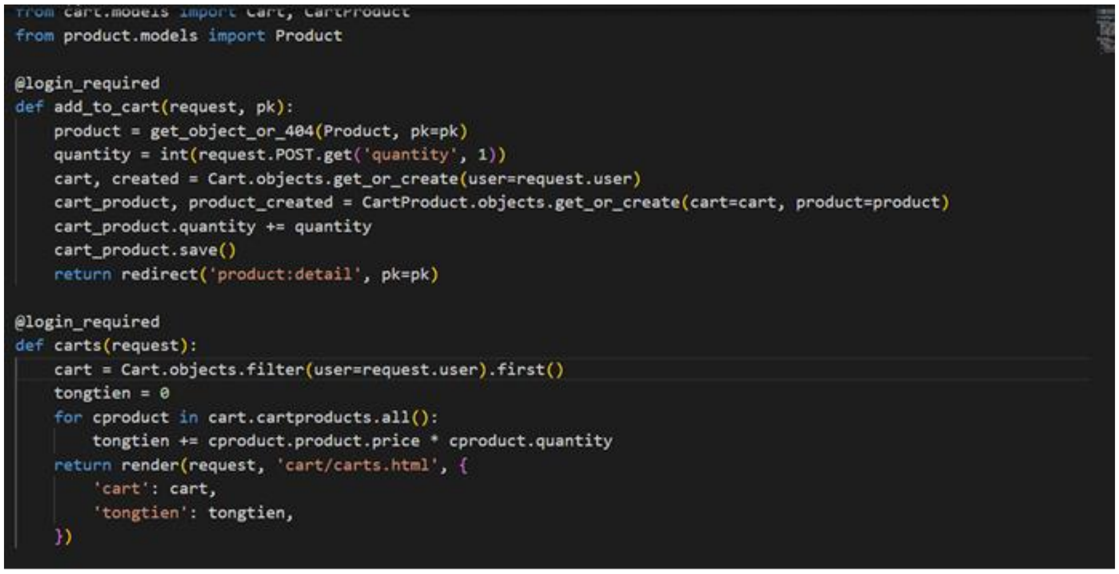
cấu hình urls tương ứng



```
from django.contrib import admin
from django.urls import path

from . import views

app_name = 'search'

urlpatterns = [
    path('', views.products, name='products'),
    path('voice/', views.productsByVoice, name='products-voice'),
]
```

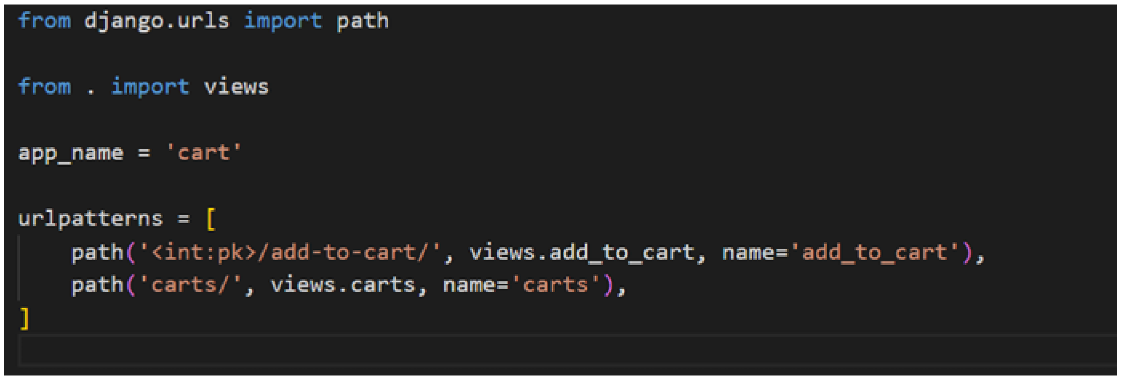
Đối với add to cart

quy trình tạo view tương ứng với add to cart và xem cart



```python
from cart.models import Cart, CartProduct
from product.models import Product

@login_required
def add_to_cart(request, pk):
    product = get_object_or_404(Product, pk=pk)
    quantity = int(request.POST.get('quantity', 1))
    cart, created = Cart.objects.get_or_create(user=request.user)
    cart_product, product_created = CartProduct.objects.get_or_create(cart=cart, product=product)
    cart_product.quantity += quantity
    cart_product.save()
    return redirect('product:detail', pk=pk)

@login_required
def carts(request):
    cart = Cart.objects.filter(user=request.user).first()
    tongtien = 0
    for cproduct in cart.cartproducts.all():
        tongtien += cproduct.product.price * cproduct.quantity
    return render(request, 'cart/carts.html', {
        'cart': cart,
        'tongtien': tongtien,
    })
```

câu hình urls



```python
from django.urls import path

from . import views

app_name = 'cart'

urlpatterns = [
    path('<int:pk>/add-to-cart/', views.add_to_cart, name='add_to_cart'),
    path('carts/', views.carts, name='carts'),
]
```